Automated Irrigation System Using Wireless Sensor Networks

Dharmasish Sahoo
Roll No.- 112EC0214

Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela
Rourkela, 769008, Sundergarh

AUTOMATED IRRIGATION SYSTEM USING WIRELESS SENSOR NETWORKS

Thesis submitted in the partial fulfillment

Of the requirements of degree of
Bachelor in Technology

in
Electronics Engineering
by

Dharmasish Sahoo
Roll No- **112EC0214**

Under the supervision of
Prof. Upendra Kumar Sahoo

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

National Institute Of Technology, Rourkela
Rourkela 769008, Sundergarh

Department of Electronics Engineering

National Institute of Technology Rourkela

Prof. Upendra Kumar Sahoo

Professor

May 16, 2016

Supervisor's Certificate

This is to certify that the work offered in the thesis entitled "***Automated irrigation system using wireless sensor networks***" submitted by ***Dharmasish Sahoo***, Roll Number ***112EC0214***, is a record of his original work carried out under my supervision and guidance in partial fulfillment of the requirements of the degree of *Bachelors of Technology* in *Electronics Engineering*. To the best of my knowledge this work has never been submitted elsewhere for any degree of any Institution in India or abroad.

Prof. Upendra Kumar Sahoo

Professor

Acknowledgment

First of all I would like to express my gratitude to my supervisor **Prof. Upendra Kumar Sahoo** for his guidance and inspiration through the entire course of this project work without whom this project would not have reached to its present form. This project has enhanced my knowledge in the field of hardware and arduino in a great deal.

Last but not the least; I would like to thank my friend **Dibya Prakash Samal** for his valuable support throughout the project.

Abstract

An automated irrigation system whose objective is to optimize water use for agricultural crops was implemented. It has a network of wireless soil-moisture and temperature sensors placed in the agricultural land. In addition to this a watering module handles sensor information, extracts required data from the string and triggers relay circuit to control pump.

The WS nodes send data to a base station which in turn shows the data received in the desktop through a MoteWorks software from Crossbow i.e. MoteView. The data received is redirected to the arduino and the required datas from the entire string is extracted. Using a code the temperature and humidity values thus obtained is compared with the threshold and accordingly irrigation is done.

Effective water administration is a noteworthy worry in numerous trimming frameworks in semiarid and parched zones. Sensor-based irrigation frameworks offer a potential answer for irrigation administration that permits makers to expand their profitability while sparing water.

Keywords: Automation, wireless sensor networks (WSNs), Crossbow

Contents

List of figures

Chapter 1

Introduction

Background and Motivation

Efficient irrigation has an important role in most agricultural cropping systems. Efficient in the sense the fields don't get over or under irrigated. But most areas of agricultural lands are effectively either over or under irrigated due to spatial inconsistency in water permeation and surplus of rainfall and irrigation, harvest water use and irrigation profundity. Less-irrigated areas are subject to water hassle, resulting in yield production loss, while over-irrigated areas go through plant disease and nutrient ooze. A possible effective solution to optimize the water organization is using a WSN controlled automated irrigation system.

A schematic flowchart of an automated irrigation system is illustrated in Fig. 1.1. The system consists of:

1) A Wireless Sensor and Base station module
2) A data acquisition module (desktop to receive the data from base station)
3) A watering module

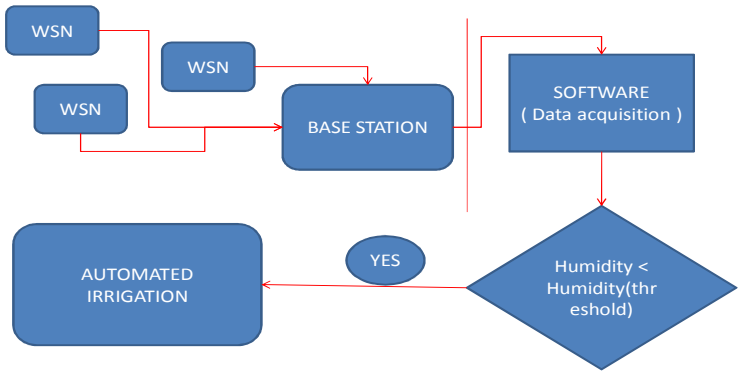

Figure 1.1: System overview

1.1 Aims and Objectives

Here we were looking to develop an automated irrigation system using wireless sensor networks. We have the wireless sensors from Crossbow which measures five readings along with temperature and humidity.

1.2 System Overview

The entire system can be broken into three modules:

1) Sensor and base station module: the sensors continuously monitor the agricultural land and send streams of data to base station wirelessly.
2) Interface b/w base station and desktop: After receiving the data at the desktop, we can see it in MoteView software.
3) Watering module: the watering unit controls the relay and motor depending on the string the arduino receives from the port emulator.

1.3 Outline of thesis

This section describes how the remaining thesis is organized. Chapter 2 tells about basics of Crossbow Sensor Kit and how to use it. In chapter 3 discussion is done about programming the WS motes and receiving data at the desktop user interface. Chapter four tells about the materials used and method implemented to carry out the hardware part of the project. Chapter five is the conclusion and also tells about future scope.

Chapter 2

Familiarization with Crossbow Sensor Kit

Wireless sensor networks empower more availability for different sensor applications that will give propelled checking, mechanization and proper control answers for the scope of development in commercial enterprises. The use of WS networks are practically boundless having numerous commercial ventures and applications having variety of innovation necessities, for example, unwavering quality, battery life, testing rate, range of frequencies, topologies, size of the system and sensor use. To address the one of a kind necessity of individual applications, Crossbow gives an expansive arrangement of remote sensor system items that permit our clients to pick the ideal answer for their industry, application and land prerequisites.

2.1 The Crossbow Starter Kit

The starter unit gives a simple and savvy answer to get direct involvement with remote sensor organizes either in the 2.4 GHz.or.868/.916 MHz ISM groups. This passage level unit gives every one of the segments expected to quickly sending a fundamental remote sensor system. The sensor hubs and entryway are preconfigured with Crossbow's dependable, self-framing, self-recuperating network organizing programming (XMesh). The MoteView application mainly for Windows PCs gives an instinctive graphical client interface to screen and deal with the remote sensor system.

By showing the network topologies, charts and graphs of the sensor data, also by configuring the sensor nodes, MoteView helps us understand the sensor data and WS network and allows easy configuration of the WS nodes.

MoteWorks:

The realization of custom Wireless sensor applications can be enabled through Crossbow’s MoteWorks software platform, which is available in a CD with the kit. MoteWorks has been specifically optimized for low_power and battery_operated networks and can provide support for the following:

• **Sensor Devices**: supports 802.15.4, OTAP and cross development tools.

• **Server Gateways**: Middleware for interfacing the wireless sensor networks with the enterprise information.

• **User Interface**: Remote analysis and monitoring Client application, also the management and config. of the WS network.

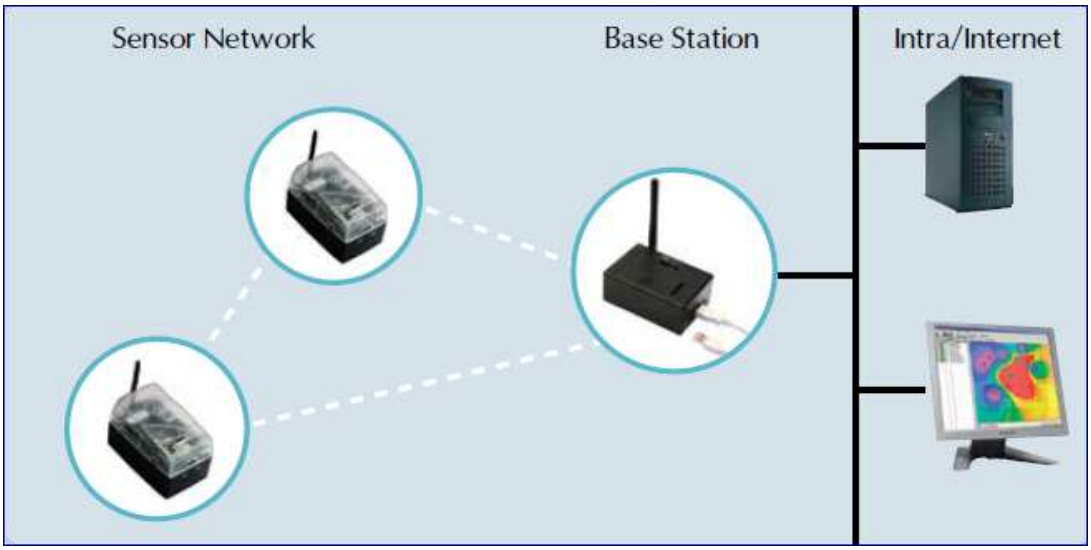

Figure 2.1: Network Architecture

2.1.1 Sensor Nodes

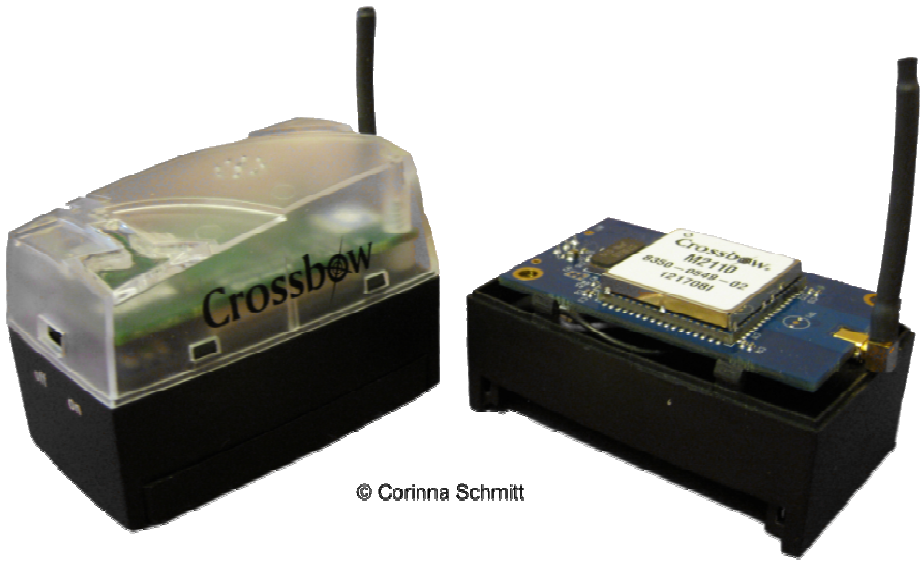

Figure 2.2: Sensor node

Specifications:

Specifications	***Range***
Relative Humidity & Temperature Sensor	
Humidity Range	0-100 % RH
Humidity resolution	0.03 % RH
Temperature range	-20˚C to +71 C
Temperature Resolution	0.01 C
Temperature Accuracy	+/- 0.5C @ 25 °C
RF Transceiver Specs.	
Frequency band	2.4 GHz
Indoor Range	30 m
Data rate	250 Kbps

Applications:

- Evaluation and development of Wireless Networks.
- Monitoring indoor environment.

2.1.2 Base Station

Crossbow's base station offers demonstrated remote innovation in a completely coordinated bundle to serve as an association between a remote sensor system and PC. The base station incorporates a processor/radio board, reception apparatus and USB interface board which is prearranged with Crossbow's solid, ad-hoc, low-control network organizing programming (XMesh) for correspondence with Crossbow's remote sensor hubs (SN24040 or SN9040).

The USB interface is utilized for information exchange between the base station and the MoteView application running on a Windows-based PC. MoteView gives an instinctive graphical client interface to picture the sensor information got from the sensor hubs and deal with the remote sensor system.

Features:

- Base Station for WSN with Pre-installed Mesh Networking Software
- PC Connection using USB
- Powered using USB
- Support for Different ISM Frequency Bands
  - 2.4 GHz, IEEE802.15.4 and 868/ 916 MHz
- Safely Packaged (indoor use)

Figure 2.3: Base Station

2.2 IRIS

IRIS is a very advanced Mote module (2.4 GHz) which is used mainly in low_power, WSN. The IRIS WSN Mote adds certain new features which increase the overall functionality and flexibility of Crossbow's WSN products.

Features:

- Up to three times improved radio range and twice the program memory over previous MICA Motes
- Outdoor line-of-sight tests have yielded ranges as far as 500 meters between nodes without amplification
- IEEE 802.15.4 compliant RF transceiver
- 2.4 to 2.48 GHz, a globally compatible ISM band
- Direct sequence spread spectrum radio which is resistant to RF interference and provides inherent data security
- 250 kbps data rate
- Supported by MoteWorks™ wireless sensor network platform for reliable, ad-hoc mesh networking
- Plug and play with Crossbow's sensor boards, data acquisition boards, gateways, and software

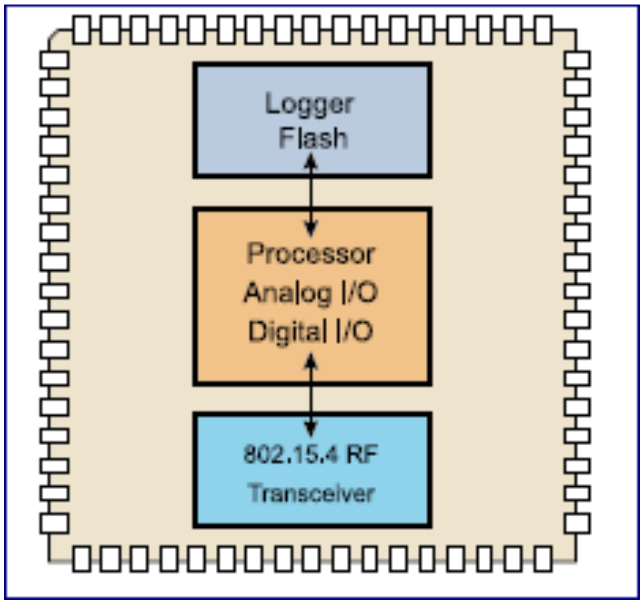

Figure 2.4: XM2110CA Block Diagram

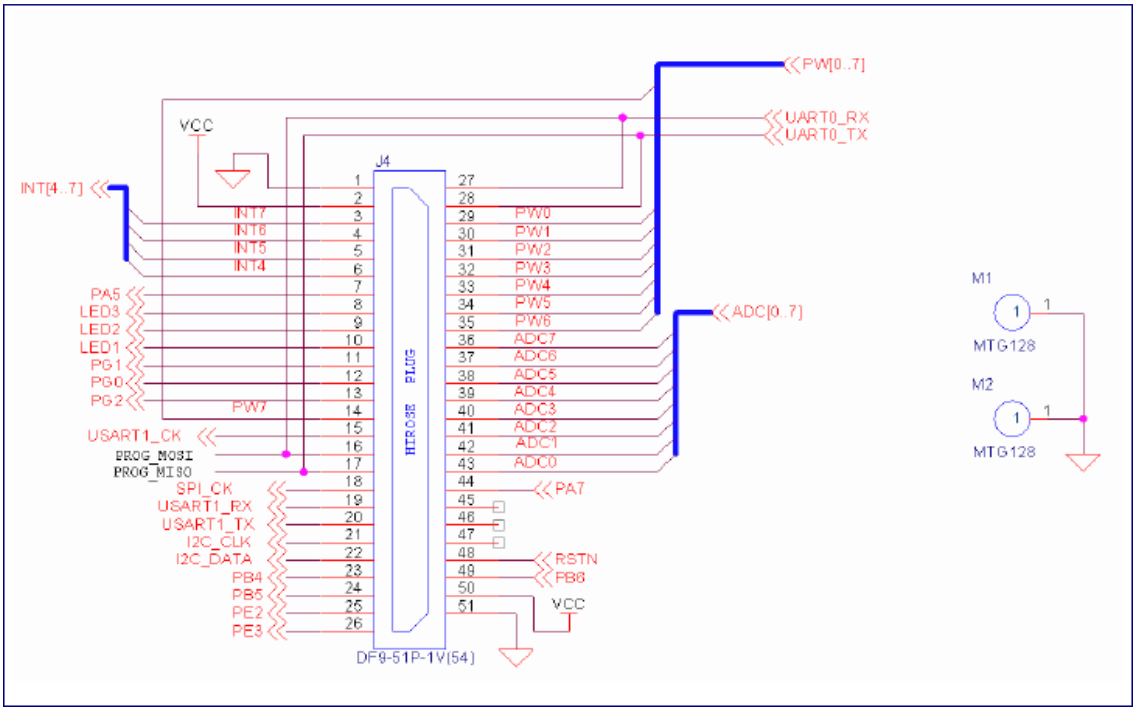

Figure 2.5: 51 pin connector

Figure 2.6: Iris Mote Bottom view

MIB 520: USB interface board

MIB 520 is used for programming the mote of both IRIS and MIcA. In fact any mote can serve as base station when mated with MIB520C usb interface board. It provides two separate ports: one for Mote programming and second one for communication while receiving from base station.

Features:

- Works as base station for WSN
- Programming the WSN motes
- Gets its power form USB

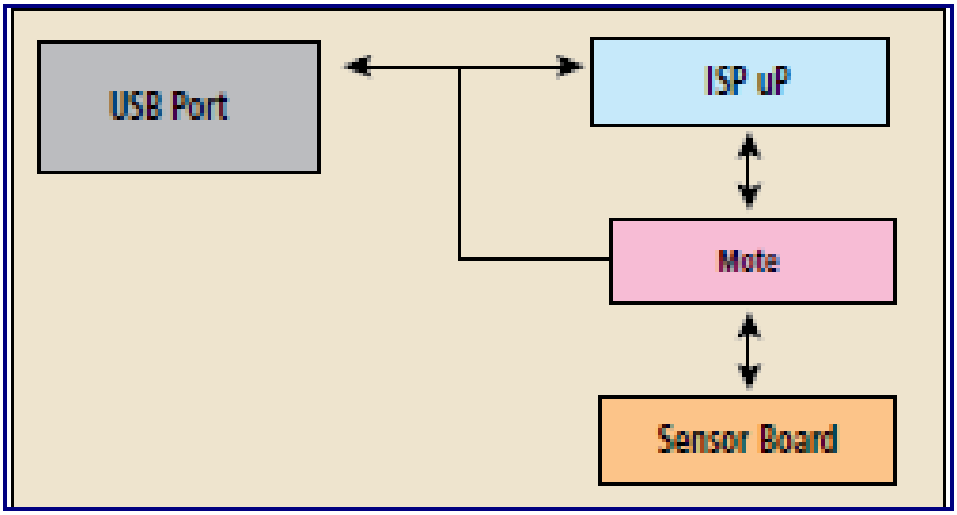

Figure 2.7: MIB520C Block diagram

Specifications:

- **USB Interface**
  - Baud Rate → 57.6 K
  - Male to Female USB cable (included with unit)
- **Mote Interface**
  - Connectors: - 51-pin
  - Indicators: - Mote LED's: Red Green, Yellow
- **Programming Interface**
  - Indicators: - LEDs → Power Ok (shows Green), Programming in Progress (shows Red)
  - Also a Switch to reset the processor and Mote.
- **Jtag Interface**
  - Connector: 10-- pin male
  - USB Bus powered

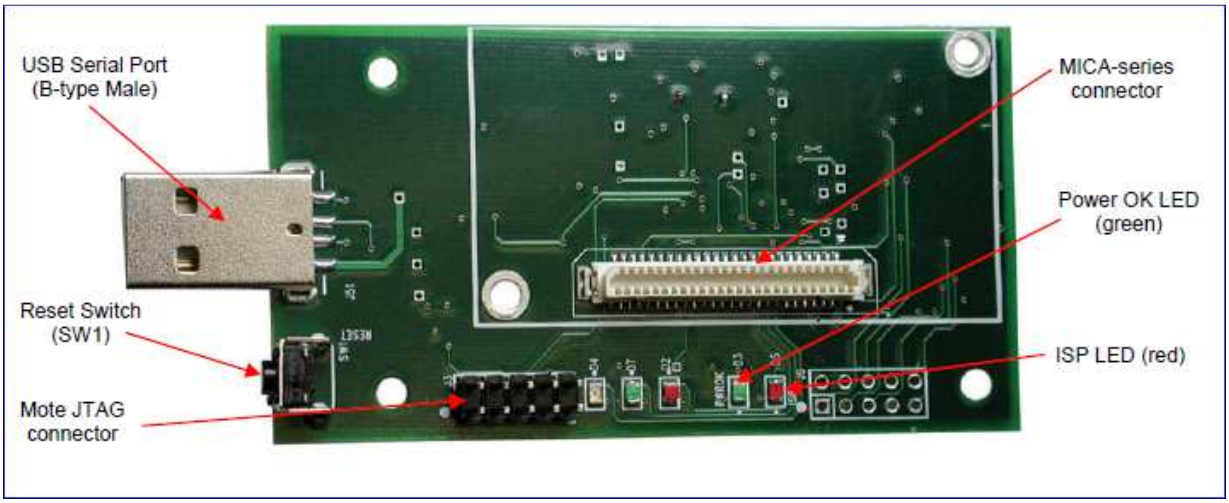

Figure 2.8: USB interface board

2.3 MoteWorks Software Platform

MoteWorks is the platform that designs softwares which program the WS motes.Different frameworks are taken into consideration for example both high power and low power motes are kept among the frameworks to be choosen from.

The two most basic software to play with the sensor and base station and for data acquisition from them are: MoteView and MoteConfig.

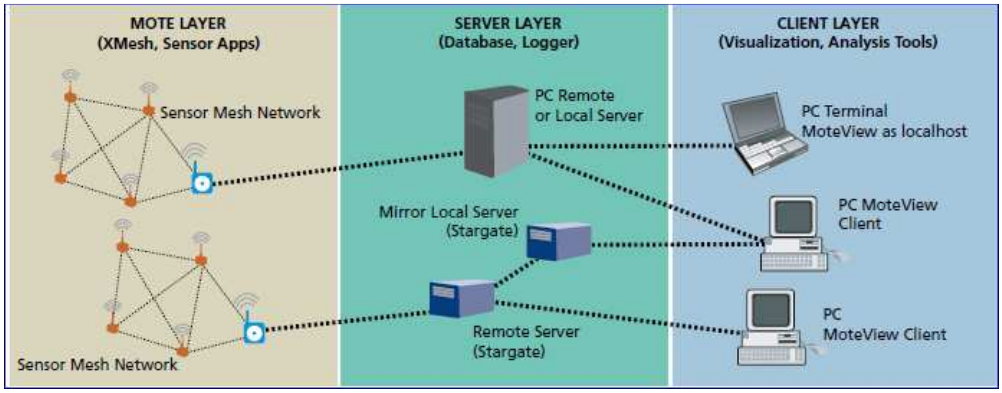

Figure 2.9: Software framework for wireless sensor network

2.3.1 MoteView

It provides analysis and visualization of data streams from multiple sensors. MoteView has its own pre-compiled firmware which helps in simple arrangement of periodic sensing applications.

2.4 MTS 400 Sensor

The MTS400 offers five common natural sensors with an additional GPS module choice (MTS420). The components offered on these sheets takes into account a wide assortment of uses extending from a basic remote climate station to a full system of natural observing hubs. Material commercial enterprises incorporate farming, mechanical, ranger service, HVAC and the sky is the limit from there. These natural sensor sheets use the most recent era of vitality proficient computerized IC-based board-mount sensors. This element gives broadened battery life where a low support, field sent, sensor hub is required.

The five different sensors in the MTS 400 are:

a) Temperature
b) Humidity
c) Pressure and temperature
d) Luminous Intensity
e) Accelerometer.

Chapter 3

Programming the Motes

3.1 "MoteConfig Software"

We have used a GUI, which predominantly works in windows XP (Server and 2000 also supported) to program the WS nodes. The motes can be programmed by loading the pre-compiled X-Mesh firmware onto the motes. We can configure the Mote id, Parent id, Group id, RF channel and RF power using this software.

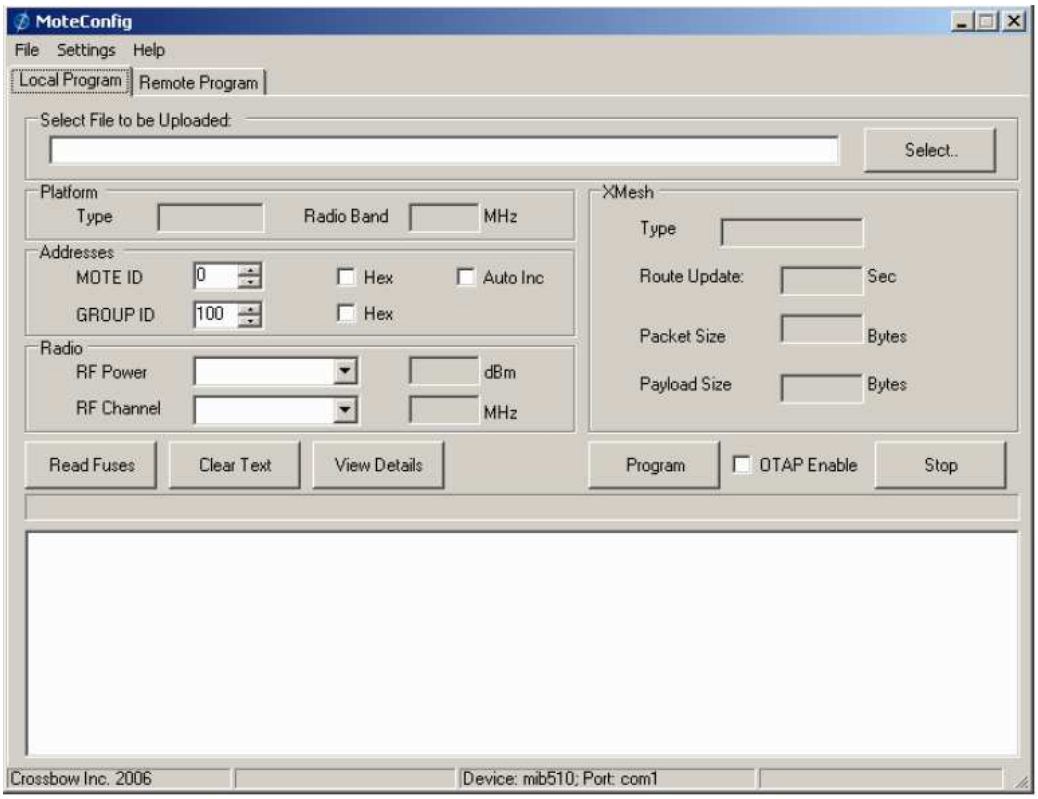

Figure 3.1: Moteconfig Application Interface

We used the MIB 520 board to program the motes.

The **MIB520** requires the installation of the *FTDI FT2232C* drivers. The MIB 520 requires an USB port for communication with the desktop. The MIB520 virtual COM port drivers will install two COM ports on the PC. The lower port is used for programming the motes and the higher port is used for communication with motes.

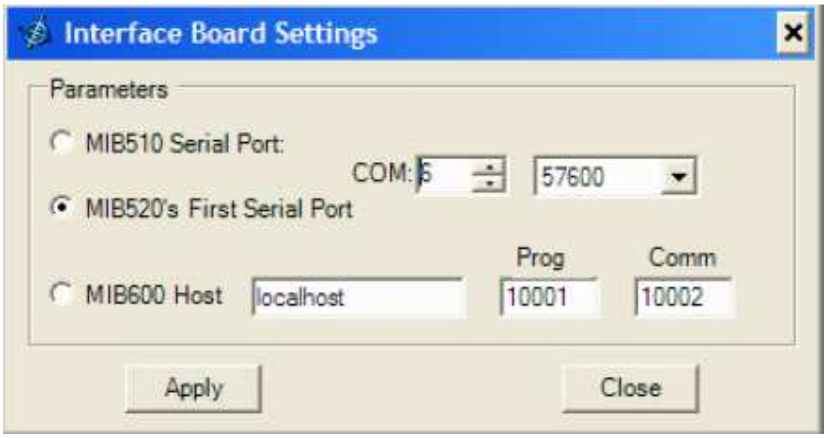

Figure 3.2: Gateway Settings

The base station Mote must be programmed with *X-MeshBase_xxx_hp.exe* and its Node-ID should be 0.

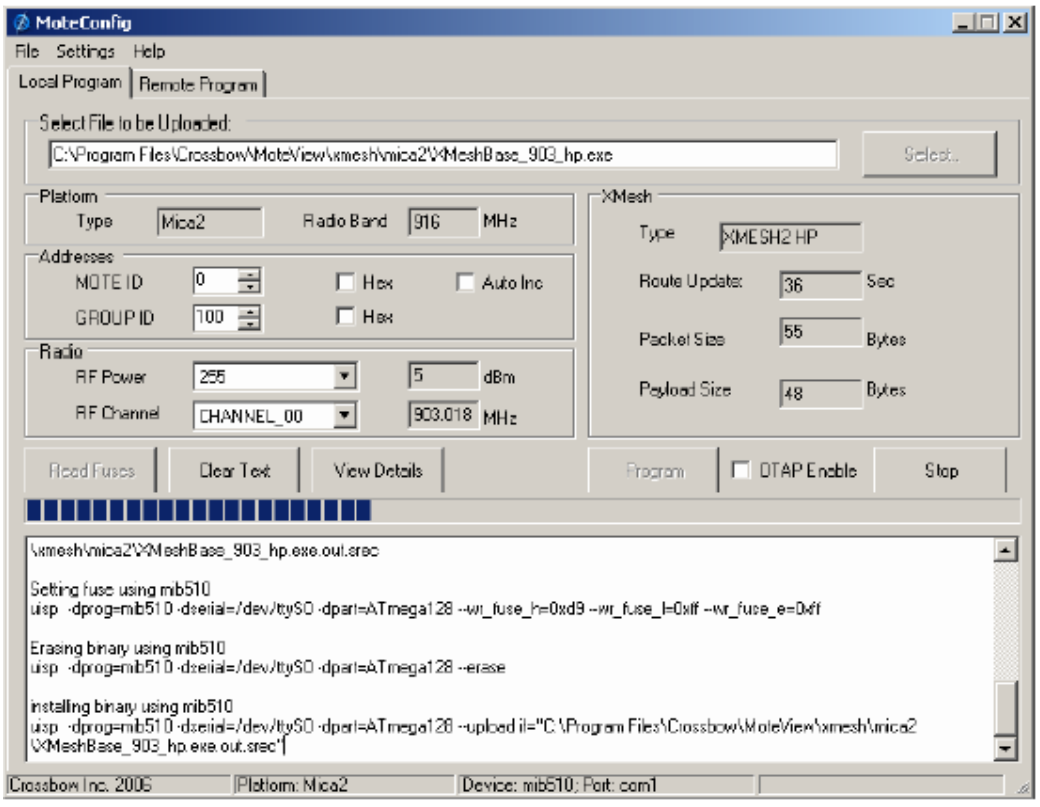

Figure 3.3: MoteConfig Programming in progress

3.1.2 Over-The-Air-Programming (OTAP)

The crossbow Moteconfig application also has this OTAP feature which allows the Motes to be reprogrammed over a wireless channel.

Also for OTAP Motes should be programmed with high_power (_hp) firmware. And the mote battery power should be higher than 2.7 V.

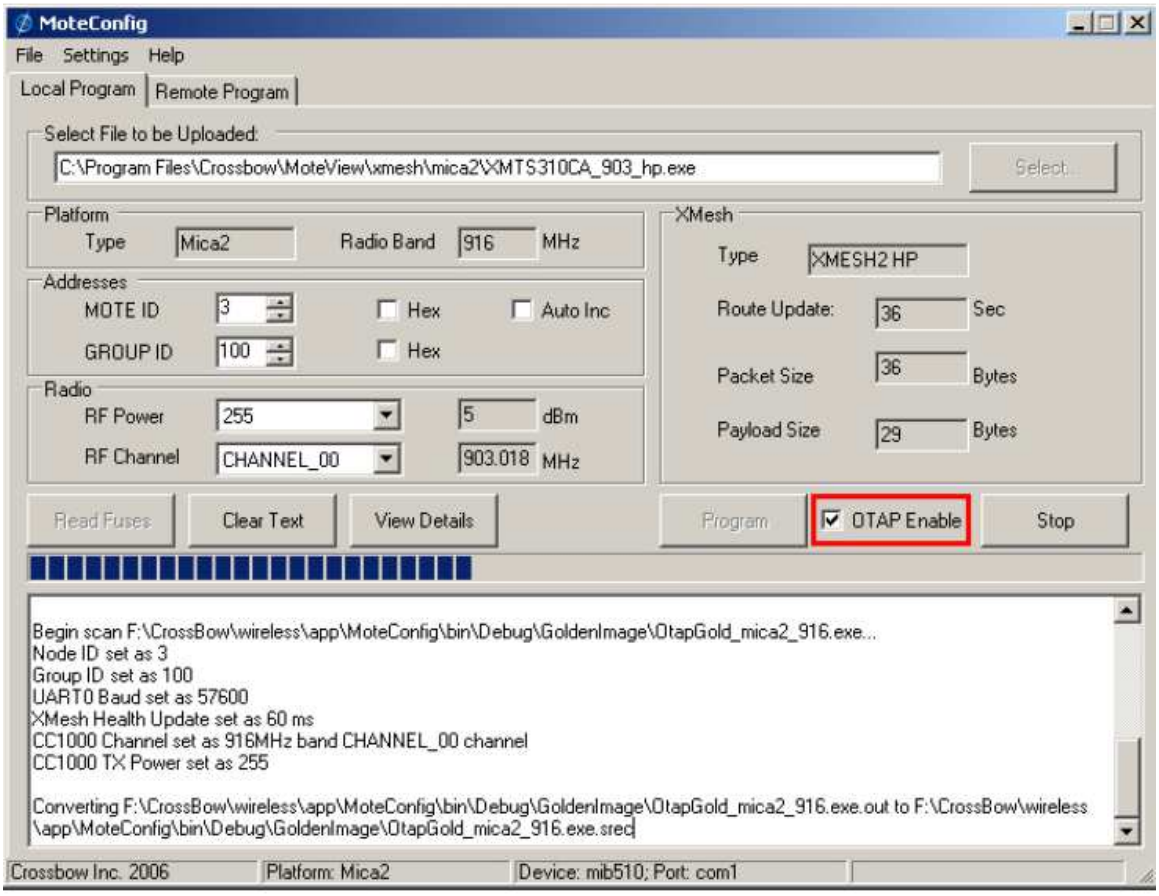

Figure 3.4: OTAP enabled Moteconfig

3.1.3 Remote Programming

Once the base station is correctly set-up it will blink with a magenta background. If it blinks periodically then it is confirmed that heartbeat packets sent by the programmed firmware are well received by the desktop.

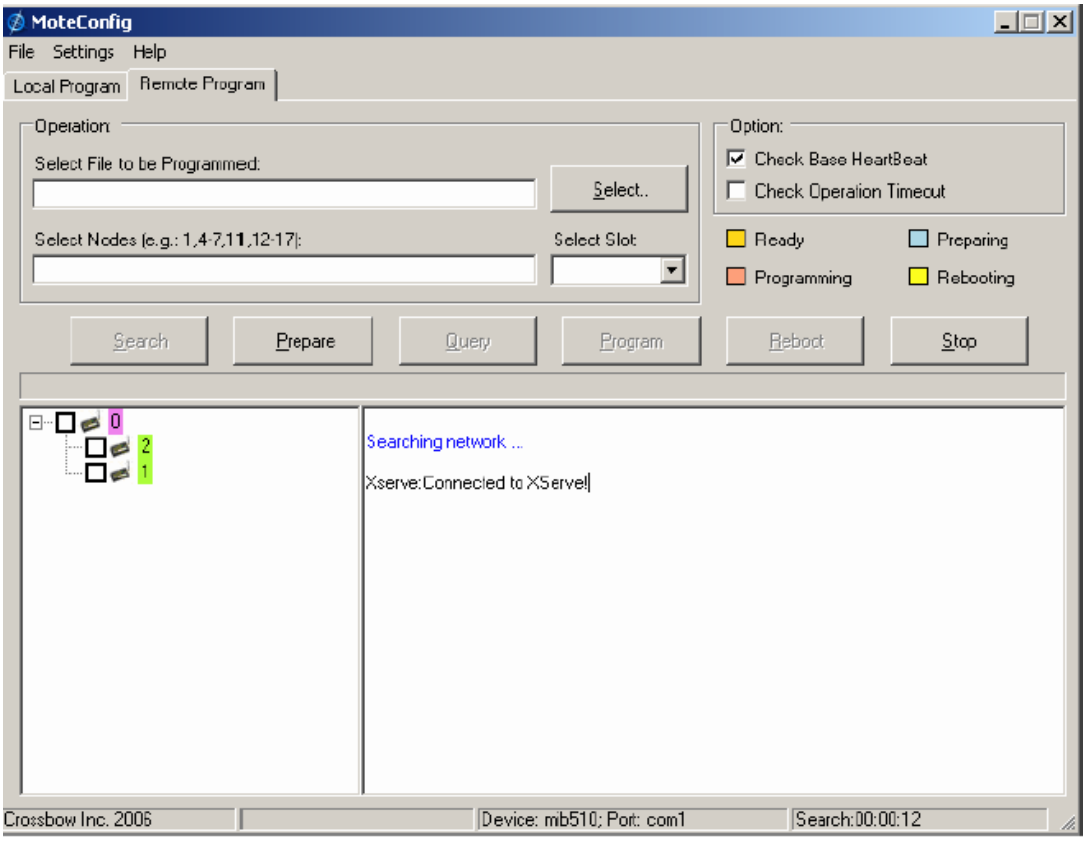

Figure 3.5: Remote programming in progress

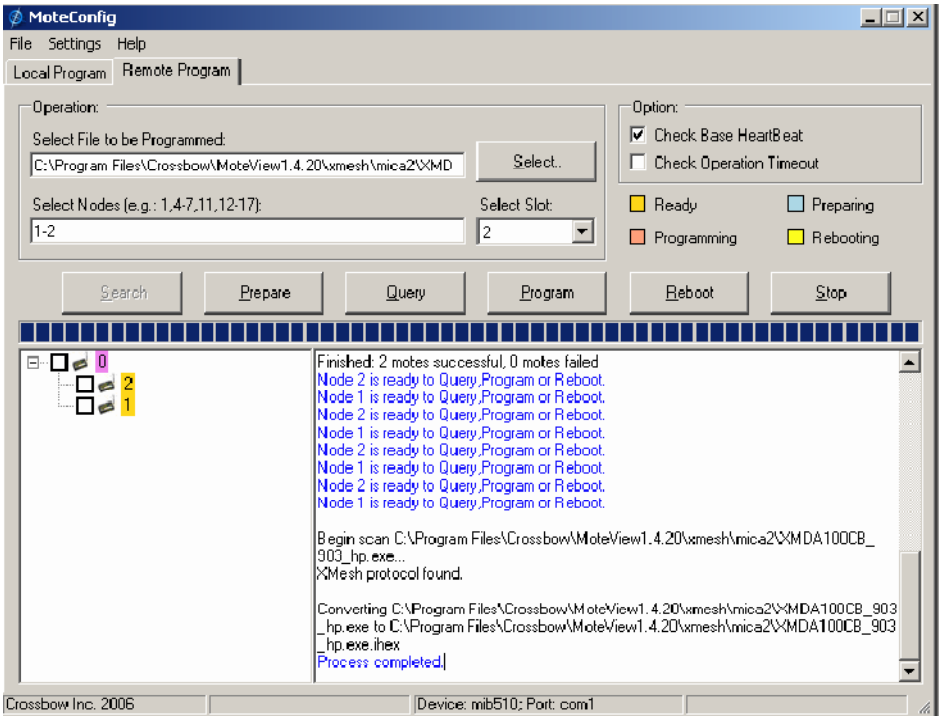

Figure 3.6: nodes in sync with OTAP

3.2 MoteView Software

MoteView is intended to be an interface ("client tier") between a client and a conveyed system of remote sensors. MoteView gives the apparatuses to improve sending and checking. It likewise makes it simple to interface with a database, to break down, and to diagram sensor readings.

Sensor and Data Acquisition Boards	Mote Platforms			
	IRIS	MICAz	MICA2	MICA2DOT
MTS101		✓	✓	
MTS300/310	✓	✓	✓	
MTS410		✓		
MTS400/MTS420	✓	✓	✓	
MTS450		✓	✓	
MTS510				✓
MDA100	✓	✓	✓	
XBW-DA100		✓		
MDA300	✓	✓	✓	
MDA320	✓	✓	✓	
XBW-DA325		✓		
MDA500				✓

Figure 3.7: sensor and data acquisition boards supported by moteview

3.2.1 Connecting to a Live WSN

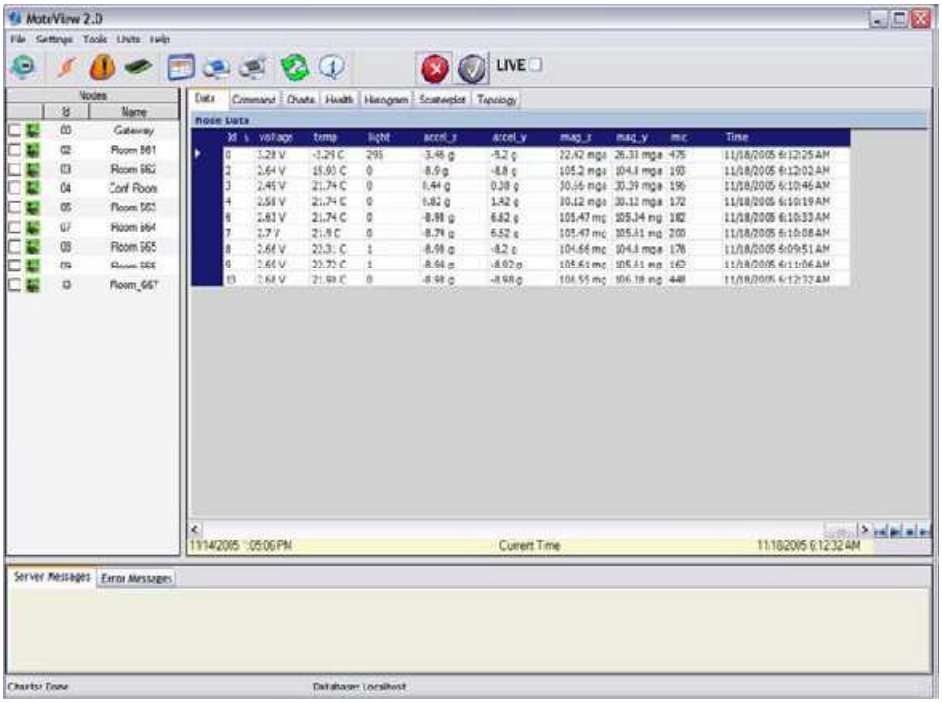

Figure: 3.8: MoteView home

Steps to get your first data on desktop form WS nodes:

1) Click connect to WSN from drop down menu

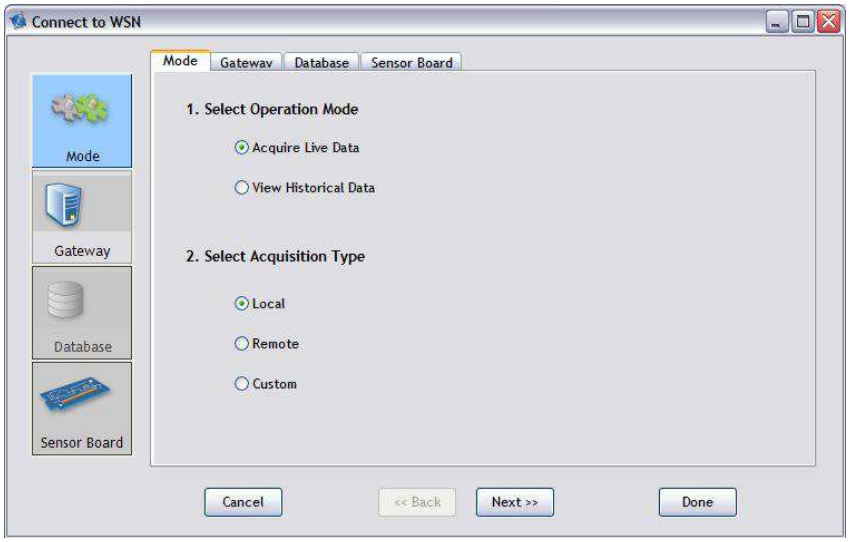

Figure 3.9: selecting operating mode

2) Select the Interface board type as MIB520 from the gateway options

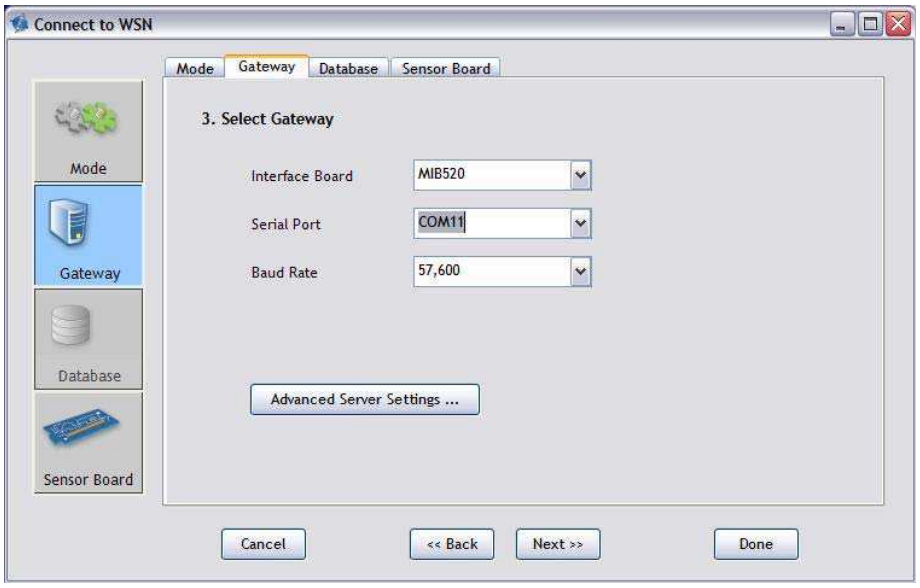

Figure: 3.10: select interface board and port

3) In the last tab i.e. **Sensor Board** tab, choose corresponding **X-Mesh Application Name** programmed into the Mote .

Figure 3.11: selecting sensor board firmware

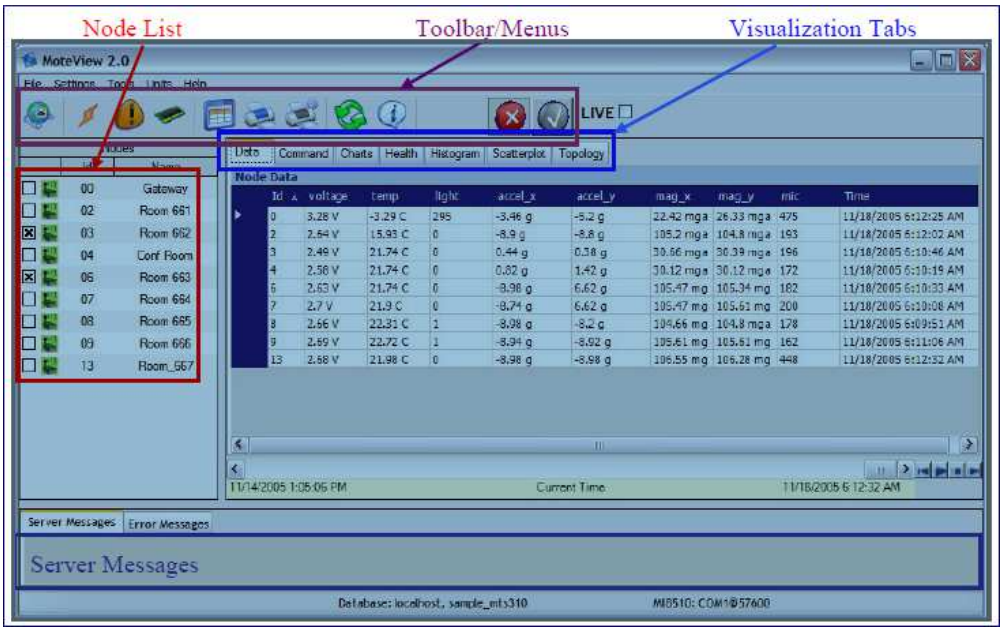

Figure 3.12: Screenshot of the Live WSN Connection

Node List Icon	Description
Gray Mote icon	No results received
Green Mote icon	Fresh results within the last 20 minutes
Moss (light green) Mote icon	Results stale by >20 minutes
Yellow Mote icon	Results stale by >40 minutes
Orange Mote icon	Results stale by >60 minutes
Red Mote icon	Results stale by more than a day

Figure 3.13: Icon properties of WS nodes

String format Sent by Wireless motes:

Data Format:

(result_time, nodeid, parent, voltage, humidity, humtemp, prtemp, pressure, taosch0, taosch1, accel_x, accel_y)

Data Received: (one data set)

(now(), 1, 0, 414, 1856, 6815, 25596, 17804, 65410, 0, 457, 463)

Chapter 4

Results and Discussions

4.1 Hardware and Software Used

4.1.1 Hardware used

- Crossbow Sensor kit
- Arduino Uno microcontroller
- Pumps – submersible
- Relay – 6V

4.1.2 Software used

- RS 232 data logger
- Com Port data Emulator

4.2 Watering Module

The irrigation is performed by controlling pumps through 40-A electromagnetic relays connected with the Arduino microcontroller.

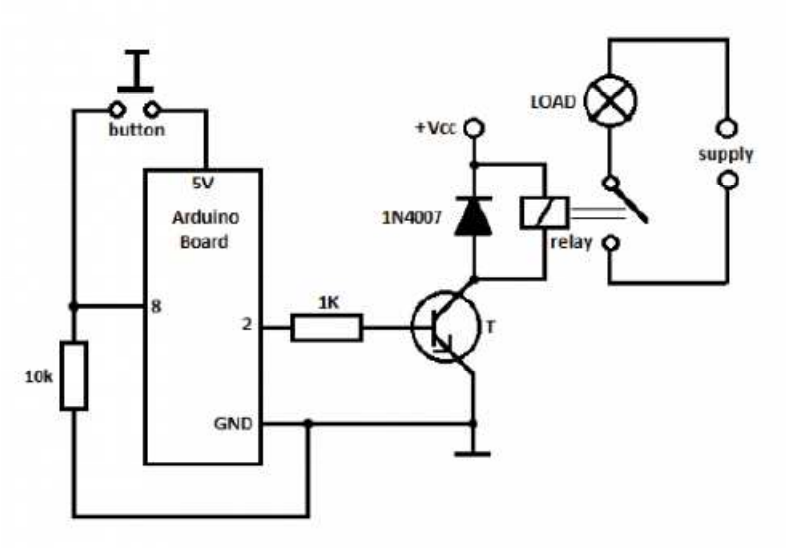

Figure 4.1: circuit for relay connection with pump

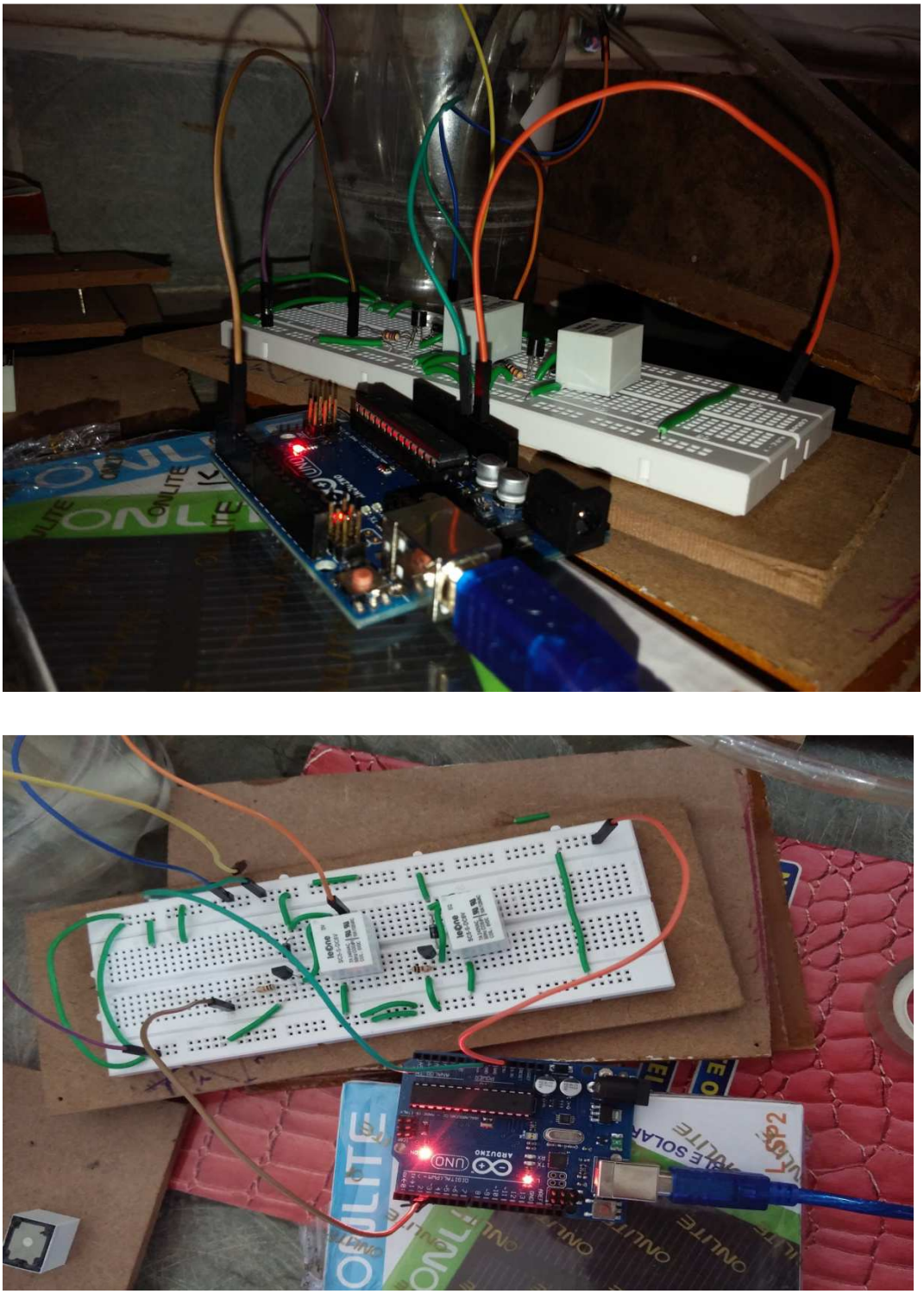

Figure 4.2: Relay connection for motor control

4.3 Interfacing Crossbow base station with Arduino

The base station sends data in a particular string format and communicates with the base station through a certain COM port.

The string format –

Data Format:

(result_time, nodeid, parent, voltage, humidity, humtemp, prtemp, pressure, taosch0, taosch1, accel_x, accel_y)

But we are only interested in the temperature and humidity values. So what we can do is redirect the data to an arduino and write a code in arduino to extract the required data values from the entire string.

But the problem is arduino is connected to the desktop through a certain COM port, obviously different from the one in which base station is connected. But arduino can do serial communication through the port in which it is connected. So the data needs to be redirected to the port to which arduino is connected.

4.3.1 Redirecting data from Base station to Arduino:

1. Using “Serial port Data Logger” and “COM Port Data Emulator”
   - Data Logger creates a log file of the data being received in a certain port.
   - COM Port Data Emulator creates a stream of data and sends it to a port
   - Use the Log file from Data Logger as a source for COM port data emulator so, data directed from Port X to Port Y.

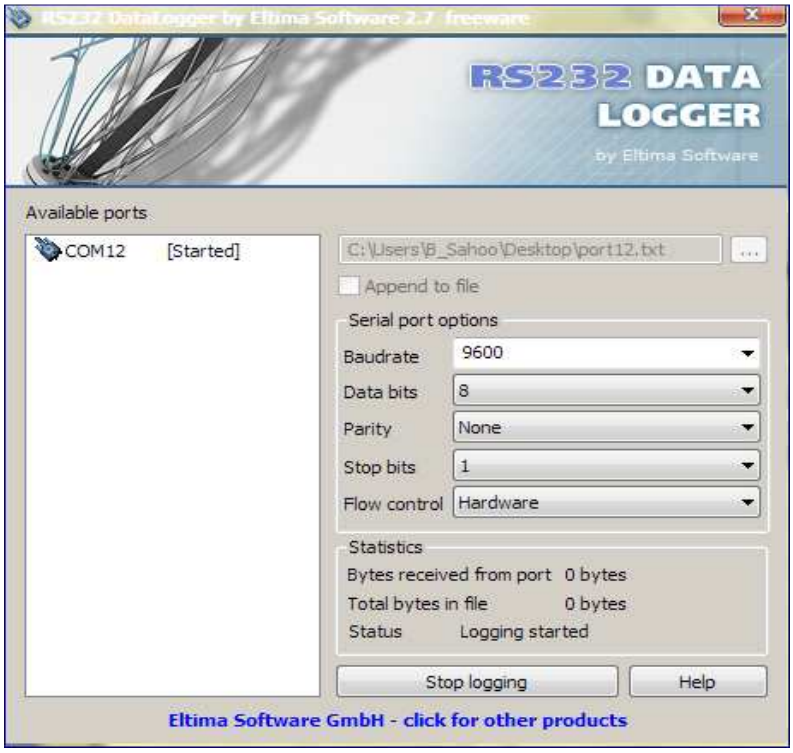

Figure 4.3: Data logger

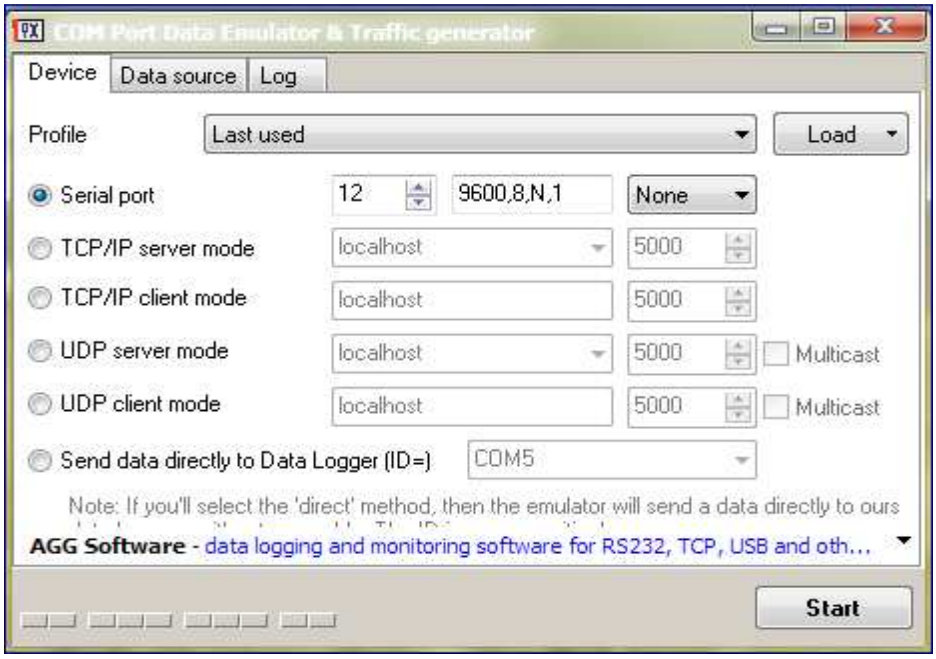

Figure 4.4: COM Port emulator

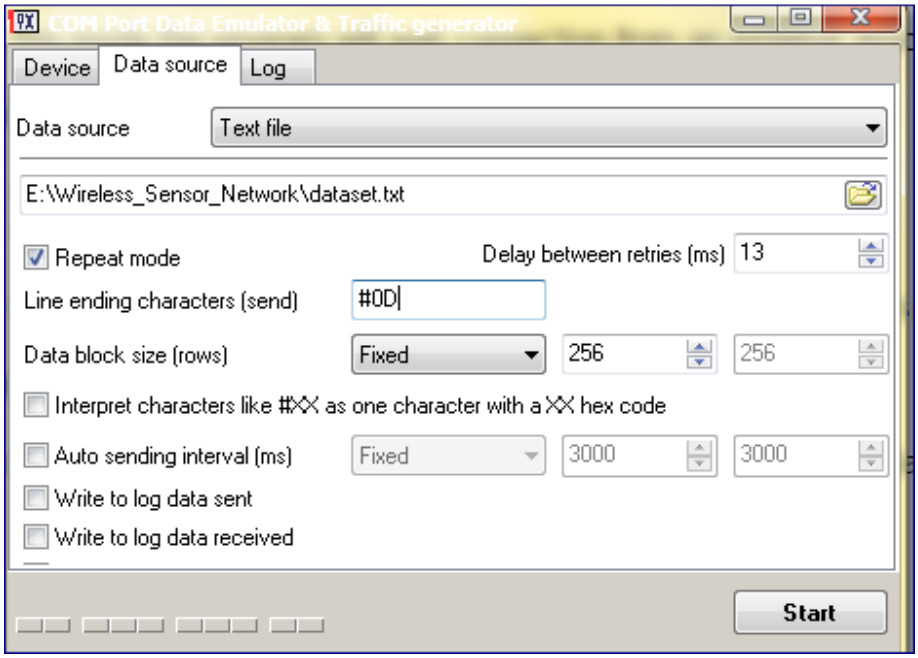

Figure 4.5: Selecting text file to read from for emulating

2. Using USB to Serial Converter
    - Direct Tx-Rx Connection B/W Base Station and Arduino

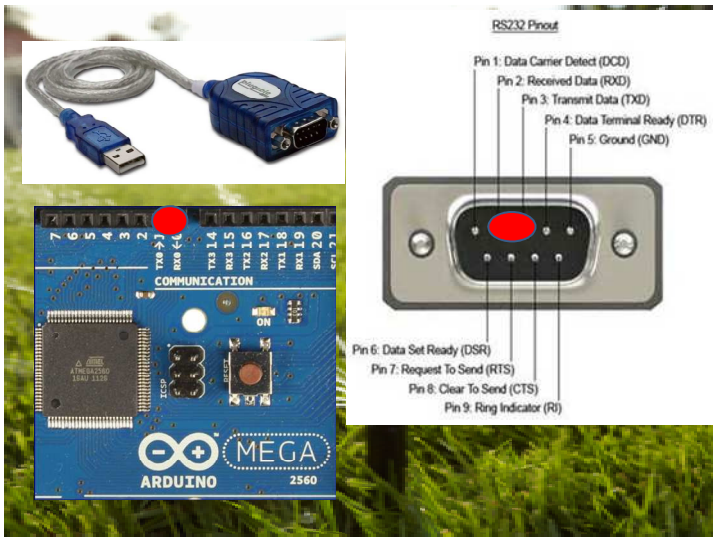

Figure 4.6: RS232 Serial pin outs and USB to female DB9 connector

Arduino code for extracting the temperature and humidity values from the string:

FINAL CODE:

```
char cRead;
int count=0; int PIN=3;
int hum_val=0; int temp_val=0; int node_id;
char humid[5]; char temp[5]; char node[1];
int i=0,j=0;
int hum_thrshld=2000; int temp_thrshld=5000;
int done=1,done_t=1;

void setup() {
// Turn the Serial Protocol ON
  Serial.begin(9600);
  Serial.println("Threshold temperature: 5000\nThreshold Humidity: 2000\n----------
------");
  pinMode(PIN, OUTPUT);
}

void loop() {
   /*  check if data has been sent from the computer: */
  if(Serial.available()) {
    /* read the most recent byte */
    cRead = Serial.read();
    if(cRead==44){
      count =count +1;
    }
    if(count==1){
      node[0]=cRead;
      node_id= atoi(node);
    }
    if(count==4){
        humid[i]=cRead;
        i=i+1;
        if(i==5)
          done=0;
    }
```

```
   if(done == 0){
     for(int i=0;i<4;i++)
       humid[i]=humid[i+1];
     humid[4]='\0';
     hum_val=atoi(humid);
     Serial.print("Current Humidity:");
     Serial.println(hum_val);
     if(hum_val < hum_thrshld)
       {
         digitalWrite(PIN,HIGH); delay(5000); digitalWrite(PIN, LOW);
         Serial.print("Humidity less than Threshold Humidity\n---Irrigation Required on sector--- ");
         Serial.println(node_id);
       }
     done=1;
   }
   if(count==5){
      temp[j]=cRead;
      j=j+1;
      if(j==5)
        done_t=0;
   }
   if(done_t==0){
     for(int i=0;i<4;i++)
       temp[i]=temp[i+1];
     temp[4]='\0';
     temp_val=atoi(temp);
     Serial.print("Current Temp:");
     Serial.println(temp_val);
     if(temp_val > temp_thrshld)
       {
         if(node_id == LOW)
           digitalWrite(node_id,HIGH);
         Serial.print("Temperature greater than Threshold temperature\n---Irrigation Required on sector--- "); Serial.println(node_id);
         delay(2000);
       }
     done_t=1;
```

```
    }
  }
}
```

4.4 Final Setup:

The final setup includes a prototype of agricultural land divided into 6 sectors. Also, it has the watering unit consisting of motor and relay connection, which in turn is controlled by the Wireless Sensors which continuously monitor the field.

Figure 4.7: prototype of agricultural field with watering unit

Figure 4.8: Wireless Sensors monitoring the agricultural land

Figure 4.9: watering being done in the required sector

Conclusion

We have successfully designed a wireless system to monitor the irrigation system using the crossbow sensor kit and the watering unit controlled by the arduino.

The Wireless automated irrigation system designed proves that water use can be reduced by a helpful percentage depending on the accuracy with which sensors respond to the environmental conditions.

The WSN irrigation system can be attuned to a diversity of crop wants and requires least amount of extra maintenance. The simple hardware constraint of the WSN automated irrigation system makes it very flexible and allows it to be scaled up for larger greenhouses or open fields.

References

[1] Joaquin Guitrezz, Villa Medina, Alejandra Nieto, Miguel Angel, "***Automated Irrigation System Using WSN and GPRS Module*** ", IEEE transactions on Instrumentation measurement, VOL .63, No. 1,January 2014.

[2] C. Gomez and J. Paradells, "***Wireless home automation networks: A survey of architectures and technologies***," *IEEE Communication Mag.*, vol. 48, no. 6, pp. 92–101, Jun. 2010.

[3] Crossbow Wireless Sensor Kit manuals.

[4] Arduino Serial Communication tutorials from "**https://www.arduino.cc/en/Reference/Serial**"